自适应Web站点设计中变色龙算法研究及实现

梅启斌[1] 白 帆[2]
(浙江万里学院EDA重点实验室 宁波31500)[1]
(华中科技大学电子与信息工程学院 武汉430074)[2]

摘 要 本文把变色龙(Chameleon)算法应用于自适应网站的设计中,通过对算法的分析具体给出了算法在聚类过程中的实现细节,并按数据挖掘的过程对自适应Web站点设计中的关键问题进行了介绍。

关键词 变色龙算法,自适应网站,Web信息挖掘,索引页面

The Research and Realization of Chameleon Algorithm Applied to Design Adaptive Web Station

MIE Qi-Bin[1] BAI Fan[2]
(EDA Key Laboratory, Zhejiang Wanli University, Ningbo 315100)[1]
(Dept. of Electronics and Information Engineering, Huazhong University of Science & Technology, Wuhan 430074)[2]

Abstract In this paper we apply Chameleon algorithm to the design of adaptive network, analyze the algorithm and descript the realization of the algorithm in clustering process in details, introduce the key design technology of Web station due to the process of data mining.

Keywords Chameleon algorithm, Adaptive web station, Web information mining, Index paper

1 引言

随着信息技术的发展,各种网站变得越来越复杂,但是它不是智能的。用户浏览网站的行为模式是动态和多样的,而大部分的网站被设计成僵化的HTML或是一些只是动态改变某些信息的站点(如asp,jsp,php等)。如何吸引更多的用户来访问网站,并为用户提供他所感兴趣的信息,成为各个Web站点设计的首要问题。于是各Web站点竞相使用基于Web挖掘的自适应站点构建技术。

自适应Web站点被定义为:能够通过学习用户的访问模式来自动改变站点的组织和表现形式的站点[1]。构建自适应网站的关键是生成索引页面(index page),索引页面是一个包括特定主题的超链接集的页面。生成索引页面的技术称为页面合成(index page synthesis)。页面合成是自动创建网页的过程,被定义为:给定站点和用户访问日志,生成包括关系密切并且与当前访问页面没有关系的超链接集页面的过程。生成不同的索引页面,也代表了不同兴趣用户群的划分。

如何对给定站点的用户群进行划分即对给定的用户进行聚类,是本文讨论的重点。在文[1]作者提出了PagerGather和Conceptual Cluster Mining算法进行聚类,考虑到以上算法的复杂性和效率,本文采用CHAMELEON算法应用于自适应Web站点建设,来进行页面综合生成索引页面,通过初步的分析取得了较为满意的效果。

2 CHAMELEON算法描述

CHAMELEON是一个在层次聚类中采用动态模型的聚类算法。在它的聚类过程中,如果两个簇间的互连性和近似度与簇内部对象间的互连性和近似度相关,则合并两个簇。基于动态模型的合并过程有利于自然的和同构的聚类的发现,而且只要定义了相似度函数就可以应用于所有类型的数据。CHAMELEON是基于对CURE和ROCK的缺点的观察:CURE及其相关的方案忽略了两个不同簇中对象间的聚集互连性的信息,而ROCK及其相关的方案强调对象间的互连性,却忽略了关于对象间近似度的信息。

图1演示了使用CHAMELEON算法进行聚类分析的过程:

CHAMELEON算法基于通常采用的K-最邻近图方法描述它的对象。K-最邻近图中的每一个点代表一个数据对象,如果一个对象是另一个对象的K最类似的对象之一,在这两个边之间存在一条边。

梅启斌 硕士,讲师,研究方向为多媒体与网络通信技术。白 帆 工学学士,研究方向为网络与通信技术。

边的权重用两个对象间的相似度表示。

CHAMELEON 通过两个簇的相对互连性(Relative Inter-Connectivity)$RI(C_i,C_j)$和相对近似(Relative Closeness)$RC(C_i,C_j)$来决定簇间的相似度：

·两个簇 C_i 和 C_j 之间的相对互联性 $RI(C_i,C_j)$ 定义为 C_i 和 C_j 之间的绝对互连性关于两个簇的内部互连性规范化，它的定义如下：

$$RI(C_i,C_j)=\frac{|EC_{(C_i,C_j)}|}{\frac{1}{2}(|EC_{C_i}|+|EC_{C_j}|)}$$

其中，$EC_{(C_i,C_j)}$是包含 C_i 和 C_j 的簇分裂为 C_i 和 C_j 的截断边；类似，EC_{C_i}(或 EC_{C_j})是它最小截断等分线的大小(即将图分为两个大致相等部分需要切断的边的加权和)。

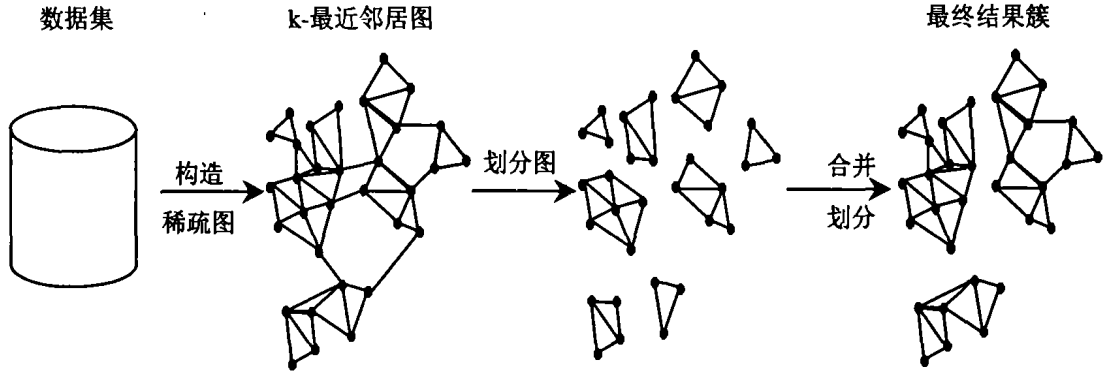

图1 CHAMELEON：基于 k-最近邻居和动态建模的层次聚类

·两个簇 C_i 和 C_j 之间的相对近似性 $RC(C_i,C_j)$ 定义为 C_i 和 C_j 之间的绝对封闭性关于两个簇的内部封闭性的规范化。它的定义如下：

$$RC(C_i,C_j)=\frac{\bar{S}_{EC(C_i,C_j)}}{\frac{|C_i|}{|C_i|+|C_j|}\bar{S}_{EC_{C_i}}+\frac{|C_j|}{|C_i|+|C_j|}\bar{S}_{EC_{C_j}}}$$

这里 $\bar{S}_{EC_{(C_i,C_j)}}$是连接 C_i 和 C_j 顶点的边的平均权重，$\bar{S}_{EC_{C_i}}$(或 $\bar{S}_{EC_{C_j}}$)是 C_i 或 C_j 的最小截断等分线的边的平均权重。

CHAMELEON 算法是一个二阶段算法：第一阶段用一个图形划分算法把 K-最邻近图划分为较小的相对独立的子簇；第二阶段用一个凝聚的层次聚类算法通过反复合并子类找到真正的结果簇。

第一阶段：生成初始子类(簇)。CHAMELEON 先构造一个 K-最邻近图 $G_k=(V,E)$，其中每一个节点 $v\in V$ 表示一个数据项。如果 v_j 是 v_i 的 k-最邻近点之一，在 v_i 和 v_j 之间就存在一条加权边 $e(v_i,v_j)$。G_k 中每条边的权重表示两个数据项之间的近似度，即两个数据项越接近，连接其边的权重越大。CHAMELEON 依据每一个递规水平在 G_k 上做最小截断，用一种图分区算法(如：hMETIS[2])将 G_k 反复划分成小的无连接子图。图 G_k 上的一个最小截断是指把 G_k 分区成两个近似的、等大小的部分，使被分区的总权重最小。然后把每一个子图看成一个初始子类，重复这个算法直至达到一定的标准。

第二阶段：算法从下向上的。CHAMELEON 通过两个类 C_i 和 C_j 的相对互连性 $RI(C_i,C_j)$ 和相对近似性 $RC(C_i,C_j)$ 来决定两个类之间的相似度。假定把一个类做最小截断时需要去掉的边的权重之和定义为该类的互连性，就可以用 C_i 和 C_j 合并后形成的类的互连性与 C_i 和 C_j 的均互连性的比率定义为 $RI(C_i,C_j)$。同样，用 C_i 和 C_j 合并后形成的类的近似度与 C_i 和 C_j 的平均内部近似度的比率定义相对近似度 $RC(C_i,C_j)$。这里，一个类的近似度是指所有做最小截断时需要去掉的边的平均权重[3]。

CHAMELEON 使用两种方法来合并相邻的簇：

1)C_i 和 C_j 的相对互连性和相对近似性必须满足用户指定的阈值 T_{RI} and T_{RC}。

CHAMELEON 计算每一个簇 C_i 和它邻近的簇 C_j 的相对互连性和相对近似性看是否满足下列条件：

$$RI(C_i,C_j)\geq T_{RI}\ AND\ RC(C_i,C_j)\geq T_{RC}$$

若一个以上的相邻簇满足上述条件，CHAMELEON 选择与 C_i 相对互连性较大的簇如 C_j 合并。重复这个过程直到没有满足条件的簇为止。

2)CHAMELEON 定义相似度函数，若 C_i 和它的相邻簇 C_j 的函数值最大则合并。函数定义为：

$$F(C_i,C_j)=RC(C_i,C_j)*RI(C_i,C_j)^{\alpha}$$

其中 α 是一个从0到1之间的用户指定的参数，减少 α 表示 $RI(C_i,C_j)$ 更重要，CHAMELEON 能自动适用类的内部特征，它能发现密度不定形状任意的聚类。若 $\alpha>1$ 则表示相对近似性更重要。重复上述过程直到没有满足条件的簇为止。

3 Web站点日志分析

Web站点日志是由网站服务器产生的不同的报告，主要包括以下几种日志：

· Access日志：记录用户访问网站的具体信息包括用户IP或解析后的域名地址、用户访问日期、请求时间、传输类型、用户访问页面的url等。

· Agent日志：可选日志，描述客户浏览器类型。

· Error日志：记录服务器执行过程中的错误日志。

· Referrer日志：可选日志，记录用户的"from-to"导航行为(例如从URL1到URL2)。

· Cookie日志：可选日志，记录访问者与服务器交互的cookie信息。

· Elf日志：扩展日志，管理员定义的任何从服务器环境获得的数据，与Access日志类似。它把以上几种日志信息合并成一行信息，目前的大部分Web服务器都支持ELF日志。本文聚类算法中的数据集来源于ELF日志。

· Application日志：基于Web的应用程序日志。该类日志由应用程序产生一般包括一些商用的或隐私的信息。

Web站点日志都是按用户的访问时间顺序记录，本文采用的Elf日志的一般结构如表1所示(Web管理员可以自定义)。

表1

日期	时间	用户IP	URI	访问时间	…
20030917	19:17:37	219.47.192.62	/~yokubota/gga/giga11.html	5	…
20030917	19:17:38	219.11.204.109	/~yokubota/index.shtml	10	…
…	…	…	…	…	…

4 在Web日志上的文档聚类形成索引页面(index page)

逻辑上，我们可以把Web看作是位于物理网络之上的一个有向图G=(N,E)[4]，其中节点集N对应于Web上的所有文档，而有向边集E则对应于节点之间的超链接。对节点集作进一步的划分，N={N_l,N_{nl}}。所有非叶节点N_{nl}是HTML文档，其中除包括文本以外，还包含了标记以指定文档的属性和内部结构，或者嵌入了超链接以表示文档间的结构关系。叶节点可以是HTML文档，也可以是其他格式的文档，如MS World、pdf等文本文件，以及图形、音频等多媒体文件。如图2所示，N中每个节点都有一个URL，其中包含了关于节点所位于的Web站点和目录路径的结构信息。

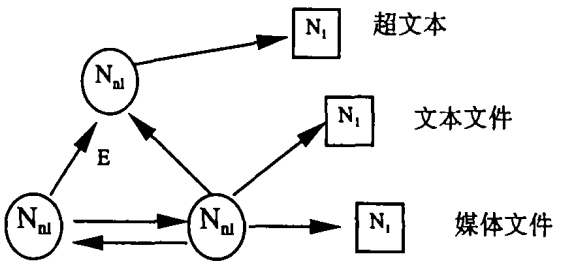

图2 Web逻辑结构图

对文档聚类形成索引页面，可分为以下5步：

Step1. 对Web日志文件的处理。

(1)在日志文件中清除由搜索引擎的Crawler以及Proxy发出的Web申请，并将其余数据装入数据库。删除日志中的图片申请，因为通常这些图片包含在某个页面中，对这些图片的申请是由HTTP协议发出的，而不是由用户发出的。

(2)抽出数据库中所有的对话过程。对于每一个用户的申请，如果相邻两个Web申请的时间间隔大于某个阈值T的话，就认为它们属于不同的会话过程。根据经验，将时间阈值定为半个小时。

Step2. 计算在滑动窗口大小内的不同Web页面间的申请同时发出的次数。建立页面间的距离矩阵。

(1)设置滑动窗口的大小。所谓滑动窗口，是指在同一个对话申请过程中，滑动窗口内的任何两个页面(P_i,P_j)申请被认为是关联的。相反，如果两个页面的申请相隔太远，则认为这两个页面不相关，不记录它们同时发出的申请。

(2)统计所有的会话过程，计算出任意两对页面(P_i,P_j)之间的同发申请的次数N_{ij}。同时，也计算每一页面单独的申请次数N_i,N_j。

(3)计算$P(P_i|P_j)=N_{ij}/N_j$。

(4)计算出页面间的距离向量矩阵。距离的定义如下：

$$D(A,B)=\sqrt{\frac{1}{P(A|B)}\cdot\frac{1}{P(B|A)}}$$

Step3. 应用CHAMELEON算法。

(1)依据距离向量矩阵，构造K-最邻近图G_k(k=3)。使用图分区算法(hMETIS[2])将G_k反复分区成许多小的无连接子图。

(2)对于任意一个子类C_i，若C_i和它的相邻簇C_j的相似度$F(C_i,C_j)$最大且满足条件则合并。重复这个过程。直到没有簇满足条件：$F(C_i,C_j)<F_T$，其

(下转第183页)

得到了一定程度的缓解，但随着网络的快速发展，老的负载均衡产品不能满足新的需要，KLBD 的设计较好地适应和满足了当前网络快速发展的需要。

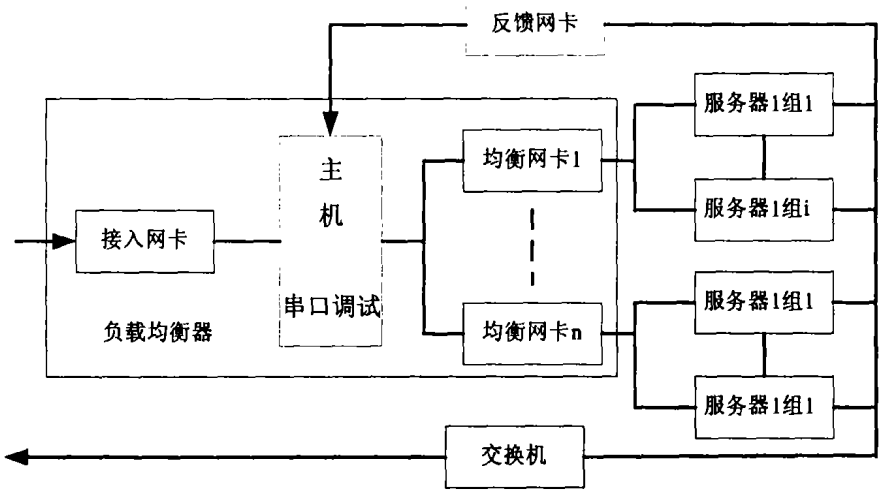

图3 基于 KLBD 的网络结构

参 考 文 献

1 王波. 使用网络地址转换实现多服务器负载均衡[Z]. http://www.lslnet.com 2000-9-23

2 李冬. 动态负载平衡 DNS 简介[Z]. http://www.lslnet.com 2000-3-31

（上接第180页）

中 F_T 是相似度的阈值。F_T 的大小影响聚类的结果，根据试验的结果选择 $F_T=0.237$ 能得出较好的聚类。

Step4. 输出聚类结果。

Step5. 根据聚类结果综合生成索引页面。

5 试验结果

本文采用的数据文件，是从网站下载的免费匿名 Web 日志。数据文件从 http://www.fin.ne.jp/~yokubota/mail-log/log.txt 处下载[5]，它记录了2003年9月30日00:05:16到16:03:13所有的访问信息。共有331个不同的访问 IP 对243个页面进行了1421次的访问。我们从中抽取出了1103次有效的会话过程。

我们在上述数据文件中使用 CHAMELEON 算法进行了聚类，表2给出了部分聚类的结果。通过对结果的观察，我们发现 CHAMELEON 算法的聚类结果是合理的相似的页面被聚集在一起。根据聚类的结果我们可以生成每一个类的索引页面。

结束语 本文就 CHAMELEON 算法应用于自适应 Web 站点的构建，给出了距离公式等算法实现的具体细节，并在此基础上针对具体的数据进行了程序实现。初步的试验表明，该方法能应用于各种不同情况的 Web 日志，它能发现不同的聚类结果，效果明显优于其它算法。

表2

C_1	/~yokubota/giga/index.html
	/~yokubota/giga/giga11.html
	/~yokubota/giga/Win98&METips&Tricks.html
	…
C_2	/~yokubota/mandsui2/mandsui2.shtml
	/~yokubota/mandsui1/mands1e.shtml
	/~yokubota/mandsui2/syou18.mid
	…
C_3	/~yokubota/jv/j-tech1.html
	/~yokubota/jv/index.html
	/~yokubota/jv/j-syxed.html
	…
…	…

参 考 文 献

1 Perkowitz M, Etzioni O. Towards Adaptive Web Sites: Conceptual Framework and Case Study. Artificial Intelligence, 2000, 118: 245~247

2 Karypis G, Han E-H, Kumar V. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. COMPUTER, 1999, 32: 68~75

3 Karypis G, Kumar V. Multilevel k-way partitioning schema for irregular graphs. Journal of Parallel and Distributed Computing, 1998, 48(1): 96~129

4 Kantardzic M. 数据挖掘：概念、模型、方法和算法. 闪四清，等译. 北京：清华大学出版社，2003

5 王晔，李德毅. 自适应 Web 站点的访问数据聚类方法. 中国人工智能进展. 北京：清华大学出版社，2001